Amendments to the Specification:

Please replace paragraph [0004] with the following amended paragraph:

[0004] In one embodiment, objects captured over a network can be queried using a graphical user interface. In one embodiment, the graphical user interface (GUI) includes a search editor to enable a user to author and edit a search that mines objects captured by a capture system. In one embodiment, the graphical user also includes a capture rule editor [[a]] to enable a user to author and edit a capture rule used by the capture system to intercept objects transmitted over a network.

Please replace paragraph [0027] with the following amended paragraph:

[0027] In Figure 1, the LAN 10 is connected to the Internet 12 via a router 20. This router 20 can be used to implement a firewall, which are widely used to give users of the LAN 10 secure access to the Internet 12 as well as to separate a company's public Web server (can be one of the servers 14) from its internal network, i.e., LAN 10. In one embodiment, any data leaving the LAN 10 towards the Internet 12 must pass through the router 12. However, there the router 20 merely forwards packets to the Internet 12. The router 20 cannot capture, ~~analyse~~ analyze, and searchably store the content contained in the forwarded packets.

Please replace paragraph [0035] with the following amended paragraph:

[0035] In one embodiment, the reassembler 36 begins a new flow upon the observation of a starting packet defined by the data transfer protocol. For a TCP/IP embodiment, the starting packet is generally referred to as the "SYN" (synchronize) packet. The flow can terminate upon observation of a finishing

packet, e.g., a "Reset" or "FIN" (no more data from sender) packet in TCP/IP. If now finishing packet is observed by the reassembler 36 within some time constraint, it can terminate the flow via a timeout mechanism.[[.]] In an embodiment using the ~~TPC~~ TCP protocol, a TCP flow contains an ordered sequence of packets that can be assembled into a contiguous data stream by the reassembler 36. Thus, in one embodiment, a flow is an ordered data stream of a single communication between a source and a destination.

Please replace paragraph [0036] with the following amended paragraph:

[0036] The ~~flown~~ flow assembled by the reassembler 36 can then be provided to a protocol demultiplexer (demux) 38. In one embodiment, the protocol demux 38 sorts assembled flows using the TCP Ports. This can include performing a speculative classification of the flow contents based on the association of well-known port numbers with specified protocols. For example, Web Hyper Text Transfer Protocol (HTTP) packets – i.e., Web traffic – are typically associated with port 80, File Transfer Protocol (FTP) packets with port 20, Kerberos authentication packets with port 88, and so on. Thus in one embodiment, the protocol demux 38 separates all the different protocols in one flow.

Please replace paragraph [0047] with the following amended paragraph:

[0047] For many of the above tag fields in Tables 1 and 2, the definition adequately describes the relational data contained by each field. For the content field, the types of content that the object can be ~~labelled~~ labeled as are numerous. Some example choices for content types (as determined, in one embodiment, by the object classification module 30) are JPEG, GIF, BMP, TIFF, PNG (for objects containing images in these various formats); Skintone (for objects containing images exposing human skin); PDF, MSWord, Excel, PowerPoint, MSOffice (for objects in these popular application formats); HTML, WebMail, SMTP, FTP (for objects captured in these transmission formats); Telnet, Rlogin, Chat (for communication conducted using these methods); GZIP, ZIP, TAR (for archives or collections of other objects); C++ Source, C Source, FORTRAN Source, Verilog Source (for source or design code authored in these high-level programming languages); C Shell, K Shell, Bash Shell (for shell program scripts); Plaintext (for otherwise unclassified textual objects); Crypto (for objects that have been encrypted or that contain cryptographic elements); Binary Unknown, ASCII Unknown, and Unknown (as catchall categories).

Please replace paragraph [0052] with the following amended paragraph:

[0052] In one embodiment, after logging on, the GUI displays an analyze view 704 to the user. In analyze view, the user can perform analysis on the objects (also sometimes referred to as documents) captured by the capture system 22. In one embodiment, the GUI also has a setup view ~~708~~730, which enables the user to control the operation of the capture system 22. The names "analyze" and "setup" are merely descriptive, and their functionalities can be given numerous

other descriptive names. For example, the analyze view 704 could be called the "research view," or the "data mining view," or any other name, so long as the view enables the user to analyze or search or mine or graph the captured objects.

Please replace paragraph [0063] with the following amended paragraph:

[0063] One embodiment of the search editor 722 illustrated in Figure 14 is configured to enable the user to create or edit a search for file transfer protocol (FTP) file transfers. The user can specify source and destination IP addresses and masks for the transfers of interest. The user can also indicate the username of the person who executed the FTP transfer. The user can also ~~proved~~ provide various transmit and receive keywords of interest in the transfer. These keywords can be indexed or non-indexed.

Please replace paragraph [0067] with the following amended paragraph:

[0067] In one embodiment, when a search is executed, i.e., run, its results – the stored objects found according to the search parameters – are displayed in a results view illustrated in Figure 16. In one embodiment, the results are listed according to various attributes, such as object type (PDF, Word, PowerPoint, Mail, ~~ect.~~etc.), content, source and destination address, size, and date captured. The results view 726 can order the results according to any such attribute selected by the user.

Please replace paragraph [0069] with the following amended paragraph:

[0069] In one embodiment, the capture rule editor 732 provides the user view tools similar to the search editor 722 as illustrated in Figure 11. The user can be provided with various keyword options (not shown) to specify words and phrases that, if they appear in the object, indicate that the object should be captured, i.e., stored[[.]], in the stored objects the search is designed to find or avoid. In one embodiment, the user can specify source and destination IP and port addresses, and masks, and protocol, to specify transmission details of the objects the user is interested in capturing. The user may also indicate the types of objects of interest by selecting from a list of possible document types.

Please replace paragraph [0071] with the following amended paragraph:

[0071] In several embodiments, the capture system 22 has been described above as a stand-alone device. However, the capture system of the present invention can be implemented on any appliance capable of capturing and ~~analysing~~ analyzing data from a network. For example, the capture system 22 described above could be implemented on one or more of the servers 14 or clients 16 shown in Figure 1. The capture system 22 can interface with the network 10 in any number of ways, including wirelessly.

Please replace paragraph [0073] with the following amended paragraph:

[0073] Thus, a capture system and a document/content registration system have been described. In the forgoing description, various specific values were given names, such as "objects," and various specific modules, such as the "registration module" and "signature database" have been described. However,

these names are merely to describe and illustrate various aspects of the present invention, and in no way limit the scope of the present invention. ~~Furthermore, various modules, such as the search engine 64 and the notification module 66 in Figure 8, can be implemented as software or hardware modules, or without dividing their functionalities into modules at all.~~ The present invention is not limited to any modular architecture either in software or in hardware, whether described above or not.